Focal-Plane Scale Space Generation with a 6T Pixel Architecture

Fernanda D. V. R. Oliveira [1]***, José Gabriel R. C. Gomes*** [1]***, Ricardo Carmona-Galán*** [2]***, Jorge Fernández-Berni*** [2]***, Ángel Rodríguez-Vázquez*** [2]

1. Universidade Federal do Rio de Janeiro, 21941-901 Rio de Janeiro, Brazil;

2. Instituto de Microelectronica de Sevilla (IMSE-CNM), CSIC-Universidad de Sevilla, 41092 Seville, Spain

Abstract

Aiming at designing a CMOS image sensor that combines high fill factor and focal-plane implementation of instrumental image processing steps, we propose a simple modification in a standard pixel architecture in order to allow for charge redistribution among neighboring pixels. As a result, averaging operations may be performed at the focal plane, and image smoothing based on Gaussian filtering may thus be implemented. By averaging neighboring pixel values, it is also possible to generate intermediate data structures that are required for the computation of Haar-like features. To show that the proposed hardware is suitable for computer vision applications, we present a system-level comparison in which the scale-invariant feature transform (SIFT) algorithm is executed twice: first, on data obtained with a classical Gaussian filtering approach, and then on data generated from the proposed approach. Preliminary schematic and extracted layout pixel simulations are also presented.

Introduction

Per-pixel pre-processing on spatial image sensor samples discards redundant data, thus decreasing the demands posed on the readout circuitry and enhancing SWaP (size, weight, and power) factors of camera systems based on these smart sensors. Image sensor architectures with embedded per-pixel processing have been proposed for a large variety of tasks [1], and their industrial exploitation is ramping up [2]. However, all these architectures share common drawbacks, namely increased pixel pitches, reduced fill factors, and lack of compatibility with advanced CIS technologies. All-in-all these drawbacks result in image quality degradation.

Based on our previous results on image sensors with per-pixel calculation of Gaussian filters, this paper presents a six-transistor image pixel that is aimed at overcoming previous drawbacks, while performing Gaussian filtering and allowing the subsequent calculation of image key-points from the filter outputs [3]. As compared to previous sensors with embedded per-pixel Gaussian filtering [4][5], this 6T pixel, which occupies $6.28 \times 6.28\ \mu m^2$ in a 110 nm CIS technology, yields 78% pitch reduction and 253% fill factor enlargement. With a two-transistor-per-pixel overhead with respect to conventional 4T pixels, the proposed image sensor architecture implements the following functions:

- image capture, which is performed in the same way as in the 4T conventional architecture;
- neighbor pixel value averaging, realized by charge redistribution;
- image smoothing, by combining pixel values inside 2×2 pixel blocks and reconfiguring the array so that a Gaussian filter approximation is performed on an image having one quarter the resolution of the original image [6];
- scale-space [7] generation, by repeatedly applying the Gaussian filter.

The proposed architecture performs simple parallel operations for the entire pixel matrix while images are captured. The processing capability of the proposed hardware can be used to optimize, in terms of processing time and power consumption, early vision tasks of image processing algorithms. The scale invariant feature transform (SIFT) [7], used for object recognition, and the Viola-Jones [8] object detector are examples of two algorithms that benefit from the presented pixel; in the first case with the scale-space generation and in the second by helping Haar-like feature computation. This paper addresses the SIFT algorithm and contextualizes the hardware-based solution in the algorithm processing flow.

The pixel architecture is explained in the next section. Then, we show how it is possible to generate the scale-space for the SIFT with the proposed hardware. By the end of the paper, system level simulations are shown, as well as schematic and layout Spectre simulations.

Proposed Pixel Architecture

In the classical 4T architecture a pinned photodiode is connected to a floating diffusion through a transfer gate transistor. When the transfer gate is activated, all the charge stored by the photodiode, which is proportional to the incident light, is sent to the floating diffusion node where it is read and sent to the output by a source follower. We propose a minor change in this architecture, as shown in Figure 1 (top). Two transistors, acting as switches, are added to the 4T architecture in order to connect the floating diffusion nodes of neighboring pixels and to create a reconfigurable array in which every pixel is connected to four of its neighbors. The sensor matrix interconnection pattern can be seen in Figure 1 (bottom). As the switches are closed, pixel averages are computed within every neighborhood in the array. As a first application of the proposed hardware, when all switches are closed an average operation is computed from the entire image, allowing an instant measure of the global luminance of the matrix, which can be used to adjust the dynamic range of the image with algorithms such as the tone mapping.

The proposed scheme permits to compute the average of pixels grouped in squares or rectangles with the size defined by the user. This simple operation allows the acceleration of Haar-like features computation, in which the mean value of neighboring rectangles within the image are compared with the goal of detecting a region where there is a high probability of finding a desired

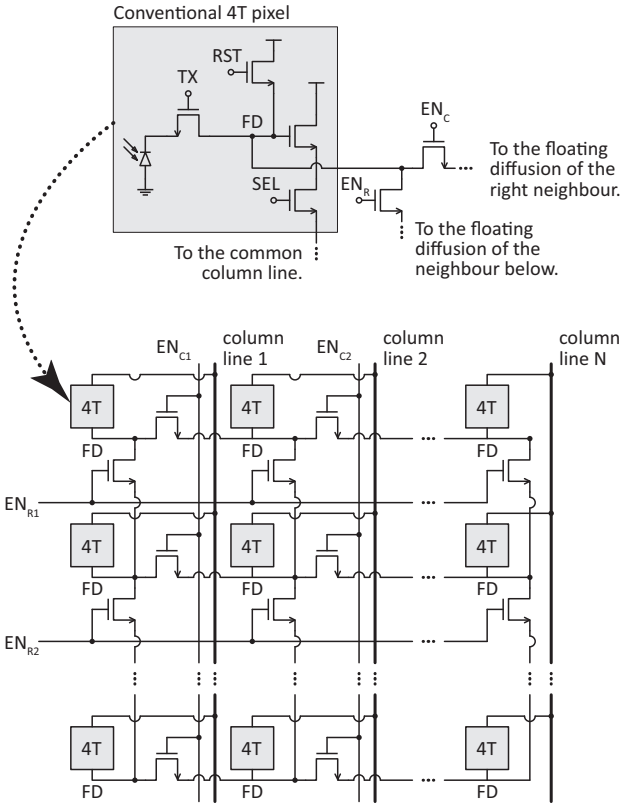

Figure 1. *Proposed 6T pixel schematic (top) and pixel matrix interconnection (bottom).*

object [8].

Filtering is achieved through charge-redistribution diffusive processes with a constant diffusion length, as explained in [6]. The pixels interconnections allow the implementation of the convolution between the 2×2 binominal kernel, shown in Equation (1), and the subsampled captured image. The central limit theorem shows that the binomial kernel can be considered a good approximation of the Gaussian filter with an equivalent standard deviation [6]. In the case of the kernel from Equation (1), it is a good approximation of the Gaussian filter with 0.5 standard deviation. The following section explains in detail the filtering process.

$$\mathbf{H} = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{1}$$

An important aspect to consider is the floating diffusion capacitance. The charge redistribution is performed using the parasitic node capacitance, which is expected to be small (in the order of magnitude of femtofarads). Increasing the floating diffusion capacitance leads to smaller conversion gain of the pixel and larger read noise, which is undesirable [9]. On the other hand, a small capacitance will be more vulnerable to charge injection and clock feedthrough errors when the charge redistribution is performed. Also, the leakage currents will have more influence if the capacitance is small. This tradeoff must be considered during the pixel design.

Focal-plane Filtering

The first step to perform the desired operation is to reduce the image resolution by computing the average values of pixels in 2×2 groups. In order to do that, odd column and row switches, as the ones presented in Figure 1, are closed ($\mathrm{EN}_{C(2n-1)}$ and $\mathrm{EN}_{R(2m-1)}$ are set to high logical values, with n and m integers, $(2n-1)$ varying from 1 to (N$-$1) and $(2m-1)$ varying from 1 to (M$-$1) for an M $\times$ N array). Once the averaging operation is performed, only one pixel needs to be sampled from each block, hence the output image has half the number of rows and half the number of columns of the original matrix.

Following this action, all switches are opened and a change of grid is set: the even column and row switches are closed (EN_{C2n} and EN_{R2n}, shown in Figure 1, are set to high logical values, with n and m integers, $(2n)$ varying from 2 to (N$-$2) and $(2m)$ varying from 2 to (M$-$2) for an M $\times$ N array). Another averaging operation is thus performed. By sampling the pixels in the same positions as the previous sample, the resulting image is equivalent to the output of the convolution between the binomial kernel of Equation (1) and the down-scaled image generated by the previous step discarding the first column and row. The result is a good approximation of the convolution between the Gaussian filter with standard deviation equal to 0.5 and the down-scaled image.

To generate the scale-space data structure, the image must be repeatedly filtered. The scale space can thus be interpreted as a pile of blurred images with an associated standard deviation each, relative to the amount of blur in the image. An additional filtering step is performed each time the configuration pattern grid is changed, always alternating the even/odd column-and-row switch patterns. The scale space must have constant distance between the images, where the distance (k) is defined as the ratio between the current image standard deviation ($\sigma_{current}$) and the previous image standard deviation at the present scale (σ_{prev}) [7]. In other words, $\mathrm{k} = \sigma_{current}/\sigma_{prev}$ must be constant. The current standard deviation can be computed as:

$$\sigma^2_{current} = \sigma^2_{prev} + \sigma^2_{filter} \tag{2}$$

In the case of the proposed focal-plane implementation we have a fixed kernel, shown in Equation (1), with standard deviation equal to 0.5, as explained in the beginning of the section. If we sample every image after applying the kernel once between one image and the next, the σ_{filter} in this case is always equal to 0.5. Consequently, the values in the standard deviation sequence, calculated with Equation (2), would be σ_1 = 0.500, σ_2 = 0.707, σ_3 = 0.866, σ_4 = 1, and so on, considering that the first σ_{prev} is zero, for the sake of simplicity, since the presented reasoning can be applied for any initial σ_{prev}. Thus, from the definition, the resulting distances in these three cases (0.707/0.500 = 1.414, 0.866/0.707 = 1.225, 1/0.866 = 1.155) are different. Nevertheless, a sequence of images with constant distance in the scale space can be obtained if we sample only the images with distance equal to 1.414 between themselves: by the definition of distance, $\sigma_{current} = \mathrm{k}\sigma_{prev}$, and plugging this into Equation (2), we find k depending on σ_{prev} and σ_{filter};

$$\mathrm{k}^2\sigma^2_{prev} = \sigma^2_{prev} + \sigma^2_{filter}, \tag{3}$$

$$\mathrm{k} = \sqrt{\sigma^2_{filter}/\sigma^2_{prev} + 1}. \tag{4}$$

If $\sigma_{filter} = \sigma_{prev}$, we have k $= \sqrt{2} = 1.414$. For the sequence of standard deviations shown above, we already have this distance for the first and second image, since σ_1 = 0.500 and σ_2 = 0.707. The necessary σ_{filter} to pass from 1 to 2 is equal to $\sigma_1 = 0.5$, resulting in k $= 1.414$. The next image from the sequence has to be generated after filtering the second image with a kernel with 0.707 standard deviation. That can be done by applying our fixed kernel twice, which leads to an equivalent σ_{filter} equal to $\sigma_{filter} = \sqrt{0.5^2 + 0.5^2} = 0.707$. The fourth image, with $\sigma_4 = 1$, is thus sampled. Consequently, the next filtering must have an equivalent σ_{filter} equal to 1, which is obtained by applying the kernel four times: $\sigma_{filter} = \sqrt{0.5^2 + 0.5^2 + 0.5^2 + 0.5^2} = 1$. The eighth image, with $\sigma_8 = 1.414$, is sampled. By continuing this reasoning we conclude that the images that must be sampled are the ones from the geometrical sequence 1, 2, 4, 8, 16, and so on. With these images, the equivalent σ_{filter} necessary to pass from one image to the next one in the sequence is equal to σ_{prev}. The sequence of standard deviations for these five selected images, for example, would be 0.500, 0.707, 1, 1.414 and 2, always yielding k $= 1.414$. The calculation of key-points is performed by computing the differences pixelwise between two successive sampled images (difference of Gaussians) and looking for extreme points across the available scales. In the presented results, the first image from the scale space is the original subsampled image.

Pixel Design and Layout

The pixel was designed using a 110 nm CIS technology. This technology, targeted for image sensors applications, although more costly than standard technologies, has the advantage of allowing color filter and lenses implementation, features higher sensitivity, low leakage current and the possibility of implementing pinned photodiodes, characteristics that improve the image quality.

Pinned photodiodes, in particular, have been widely used in the digital camera market due to their low noise, high quantum efficiency and low dark current [9]. The pinned photodiode is being used with the goal of improving the image quality and leveraging the technology full potential. The simulations presented in the next sections aim at representing the pixel behavior after the charge transfer between the pinned photodiode and the floating diffusion.

The main aspects considered during design were the floating diffusion capacitance and the minimum desired fill factor. The three switches in the pixel (the select switch and the two new switches) have minimum size allowed by the technology: width equal to 180 nm and length equal to 340 nm. Consequently, they have small capacitance, thereby reducing the charge injection. The reset transistor has 1 µm width and 340 nm length so that the voltage drop across the transistor is low when the reset is set on. The transfer gate transistor also has 1 µm width in order to reduce the bottleneck effect during the charge transfer, and length equal to 450 nm, which is the minimum length defined by the technology for transfer gate transistors. The source follower is the largest transistor, featuring 4 µm width and 340 nm length. These values have the goal of increasing the floating diffusion capacitance, since this transistor is the one with highest influence in this capacitance.

The pixel layout is shown in Figure 2. The technology allows the use of four metal layers. The pixel has a 3 µm × 3 µm sensing area, and a 6.28 µm × 6.28 µm full area, resulting in a fill-factor of 22.8%. The reset source voltage was included in the layout in a separate line aiming at assuring flexibility in a future experimental test. Since we found no available information on models for the pinned photodiode, it is also in our interest to investigate its behavior. A technique for measuring pinned photodiode and transfer gate parameters that involves reducing the reset voltage during the integration period is presented in [10]. This type of measure may help the design of future chips and the better understanding of this chip. For the electrical simulations the reset voltage was connected to the source voltage V_{DD} equal to 3.3 V.

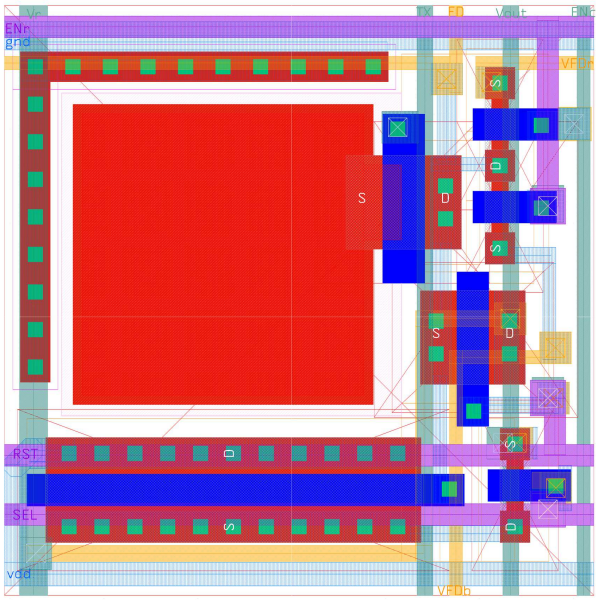

Figure 2. *Proposed pixel layout*

In order to evaluate the additional hardware influence on the output signal, Silvaco process and simulation softwares (ATHENA and ATLAS frameworks) are being considered for studying the photodiode physical properties [11]. It might also help improving the pixel design.

Image Processing Algorithms

Gaussian filtering is extensively used in image processing algorithms. It has the property of smoothing the image, which can be used to reduce noise. Another application of this operation is to generate the scale space of an image, created through successive Gaussian filtering steps [7]. The scale space guarantees the scalability of object recognition algorithms, so that a desired object can be recognized across different scales.

SIFT (scale invariant feature transform algorithm) has been widely used and explored lately for object recognition. The scale space is used to find stable key-points from a given image through different scales. These key-points are considered to be unique and may be used to describe the image, assuring the algorithms reliability and scale invariance. This algorithm presents very good results in the object recognition field, but it also requires a huge amount of computational load [12].

For object detection, the Viola-Jones algorithm is a very good alternative because it has a simple flow and a high accuracy rate. The Viola-Jones is a fast algorithm, but it is still interesting to reduce its processing time and power for embedded systems. For applications in which only the object region is desired, time and bandwidth savings can be achieved by encoding only the part of the image where the object was found. Focal plane processing can help by reducing the amount of hardware outside the pixel matrix, and consequently reducing the power consumption necessary to identify a object.

The proposed architecture may help both algorithms by saving time and power processing. In the following sections, both the SIFT and the Viola-Jones steps are explained in more details.

Scale Invariant Feature Transform

This algorithm [7] uses image descriptors to recognize objects and compare scenes. It may be used for object recognition, tracking, panoramic assembling and 3D modeling, among others. It searches the image for characteristic points, represents them using vectors and compares these vectors with vectors from other image in order to find correspondences.

The first step of the SIFT is the scale space generation: Gaussian kernels with different standard deviations, carefully chosen with the goal of generating a linear scale space (constant distance between images), are used to filter the image. After filtering the image a certain number of times (depending on the standard deviation of the filter) the image must be subsampled and the filtering process is repeated [7]. Each time the image is subsampled a new octave is formed. Thus, the scale space is divided by octaves, which are a set of images with same resolution and increasing smoothing. The following step is the Difference of Gaussian (DoG) computation, which is a good approximation of the Laplacian of Gaussian (LoG). It produces stable points of interest. For each point of interest an orientation is calculated according to its gradient. The points of interest are then described by histograms of angles calculated using the main orientation as a reference.

Although the scale space generated by the chip is different from the one proposed by Lowe [7] in terms of kernel size, image distance across scales and initial standard deviation, the benchmark results and the proposed hardware results are comparable, as it will be shown in the Results section. System-level simulations are being performed in C based on the OpenCV SIFT implementation. The algorithm scale space computation under investigation is fully compatible with the proposed hardware.

Viola-Jones Object Detection Algorithm

The Viola-Jones algorithm was first introduced with the goal of being a simple and efficient algorithm for face detection. Furthermore, it also presents high efficiency when trained for other objects, such as car and pedestrian [8]. Lately, it has been largely used worldwide. A cascade of increasingly complex classifiers that use Haar-like features is employed to find regions with a high probability of detecting the desired object. Each classifier has a low false negative rate but also a high false positive rate, which means that when a classifier discards a window it has high probability of not having the object, but when it accepts one it does not mean that it is necessarily the targeted object. When the classifiers are put together in a cascade the overall false positive rate decreases, resulting in a high accuracy rate for the cascade [8].

A window that slides through the image is used to perform the analysis by searching for the targeted objects with the size of the window. For the sake of scale invariance, after each search through the entire image the window and feature sizes increases, so that objects can be found at different scales. That means that the same algorithm is performed repeatedly until the search is held at all the desired scales.

Luminance sensitivity is controlled by normalizing the pixels inside the search window. The features are calculated and compared to a threshold that considers this normalization. In the cascade of classifiers, if the features from the first classifier satisfy the requirement, new features, from the subsequent classifier are computed and compared to another threshold. The first classifier needs very little processing and is responsible for eliminating a large number of windows not containing the targeted object. The last, and more complex classifier, is only computed for a few windows, the ones that have higher probability of containing the object. In order to make the computation of the features faster, the integral image concept is introduced. In an integral image, each pixel is defined as the sum of the pixels from above and to the left of its position and, from the integral image, the sum of each rectangle from the Haar-like features can be computed with a maximum of four operations.

Although the Viola-Jones algorithm is very simple and the computation of the integral image helps to speed it up, it still demands significant computational resources, specially for embedded circuits applications. The pixel architecture proposed, which is able to compute mean values, may help by computing the mean of rectangles from the first features in an analog way, resulting in a reduction on the amount of windows going through the entire algorithm [13]. To generate a classifier cascade that takes hardware limitations into account, further study on Viola-Jones training is required.

Results

The proposed hardware is able to generate a scale space with parameters that are different from the one initially proposed for the SIFT algorithm. This section shows, by means of repeatability comparisons, that the scale space generated by the chip is suitable for SIFT applications. System level simulation results are presented.

Spectre simulations from Cadence are also presented for both the schematic and extracted layout. These simulations aim at identifying the error magnitude after the charge redistribution, by describing how the voltage in the floating diffusion changes during the operation.

System Level Simulations

Repeatability is an important parameter that is used to evaluate the key-points stability. It compares the key-points that are found in two images describing the same scene, but transformed in basic aspects such as blur, viewpoint and rotation, among others. Most of the key-points must be found in both images. By comparing the repeatability of the key-points found in the scale space generated by the proposed method with the repeatability of the key-points found in the scale space proposed by Lowe, we are able to evaluate the proposed method quantitatively.

Given two images I_1 and I_2 representing the same planar same scene, the key-points (x_1) found in I_1 can be related to

the key-points (x_2) found in I_2 by the homography matrix H_{1to2}, where x_2=$H_{1to2}x_1$. Depending on the image change, some points might be occluded and not appear in both images. These points cannot be considered in the repeatability measure [14]. The key-points that are present in both images can be defined as:

$$\widetilde{x}_1 = \{x_1 | H_{1to2}x_1 \in I_2\}; \qquad \widetilde{x}_2 = \{x_2 | H_{2to1}x_2 \in I_1\}. \tag{5}$$

To compute repeatability, it is also considered that there is an uncertainty regarding where the key-point can be found after applying the homorgaphy. A neighborhood ϵ is thus defined and the set of matching key-points ($\widetilde{x}_1$, $\widetilde{x}_2$) is composed by points separated by a distance not larger than ϵ after being multiplied by the corresponding homography:

$$R_2(\epsilon) = \{(\widetilde{x}_1, \widetilde{x}_2) | dist(H_{1to2}\widetilde{x}_1, \widetilde{x}_2) < \epsilon\}. \tag{6}$$

Considering the above described aspects, the repeatability rate ($r(\epsilon)$) for image I_2 is defined in [14] as:

$$r(\epsilon) = \frac{|R_2(\epsilon)|}{min(|\{\widetilde{x}_1\}|, |\{\widetilde{x}_2\}|)}, \tag{7}$$

which is the ratio between the number of corresponding key-points found considering a neighborhood (ϵ) and the minimum number of key-points that can be found.

OpenCV provides a set of SIFT functions [15][16] that are reliable for generating the key-points as Lowe proposed and for performing the comparison by computing the repeatability. The database has eight different images with five transformations each (H1to2 until H1to6), including blur, viewpoint, zoom, rotation, illumination change and JPEG compression [17][18]. Pairs of images with their respective homography matrix and key-points are used in the repeatability computation.

Lowe's scale space has initial standard deviation equal to 1.6, three scales per octave and the number of octaves computed according to the images resolution. The following steps are the computation of the difference of Gaussians and key-points search. All these steps were performed with OpenCV functions. Although an image upscale is also proposed in [7], it was not implemented in order to guarantee a fair comparison between Lowe's and the chip scale space. A method for performing the resolution increase after the generation of the scale space pyramid is proposed in [19].

The chip scale space method described in the 'Focal-plane Filtering' section was implemented in C. The resulting images with constant scale distance equal to 1.414 were used for the difference of Gaussians. The search for key-points was also performed by an OpenCV function. Repeatability results are presented in the table 'System Level Simulation Results'. The repeatability associated with the scale space computation as originally proposed by Lowe (but without increasing image resolution) is shown in column 'Original Method', and the repeatability associated with our method is shown in column 'Proposed Method'. The table also presents results when a random noise is added in the chip method, which will be analyzed in the next section.

The transformation matrix is different for each image: images 'bikes' and 'trees' have blur transformations; 'graf' and 'wall' have viewpoint transformations; 'bark' and 'boat' have zoom and rotation transformations; 'leuven' has illumination change; and 'ucb' has JPEG compression. As can be seen in the table, in most cases the repeatability values of Lowe's method and the (proposed) chip method are very similar. The chip presents a low repeatability, though, when zoom transformations are applied (images 'bark' and 'boat') because only three octaves where implemented for the chip method. Further study on the number of scales is important because current leakage may limit the number of charge redistributions. The results nevertheless show that the method proposed is a valid alternative for the scale space generation, since the average repeatability considering all images is only 3% lower than Lowe's.

OpenCV SIFT implementation uses two thresholds to filter weak features, namely contrast threshold and edge threshold. The first one filters key-points found in low-contrast regions and the second filters edge-like key-points. These are undesirable key-points characteristics because features generated under these conditions may be unstable and, as a consequence, it becomes difficult to find correspondence when a transformation is applied to the image [20]. These thresholds where not optimized for the hardware scale space method, so it is expected that the results can be improved by tuning these values to the desired scale space.

Schematic and Layout Simulations

In order to understand the error in the averaging operation that is due to the charge redistribution process, simulations were performed for one row and for one column of four pixels each. The charge redistribution, in both cases, is performed between two neighboring pixels. The pixels from the borders were added to guarantee that the floating diffusions have the same number of transistors connected and, consequently, have the same equivalent capacitance.

For these simulations an ideal charge transfer between the photodiode and the floating diffusion was considered. Figure 3 shows the model used for the pinned photodiode and transfer gate. The idea is to transfer all the charge stored in the capacitor to the floating diffusion when the transfer gate switch is closed. A buffer is used to assure that the capacitor will have the same voltage at both terminals when the switches close, and that all the current generated when that happens will go to the floating diffusion.

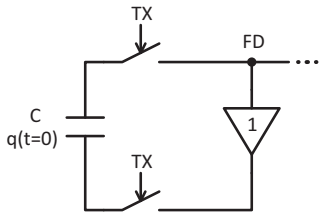

Figure 3. *Pinned photodiode and transfer gate ideal model used for simulations.*

The control signals necessary to perform the simulations are presented in dotted lines in Figures 4 and 5. For both the line and column simulations we show the RST, whish is responsible for setting the floating diffusion voltage, and the TX, which transfers the photodiode accumulated charge to the floating diffusion, thus reducing its voltage. The next signal is the charge redistribution enable. When two pixels are in the same column (Figures 4 and 5 top) two rows are connected together, so the EN_R signal is activated. When the pixels belong to the same row (Figures 4

System Level Simulations Results

Image	Repeatability		
Bark	Original Method	Proposed Method	Proposed Method with Noise
H1to2	58.55%	61.01%	57.14%
H1to3	62.02%	27.30%	26.49%
H1to4	63.93%	23.01%	30.43%
H1to5	54.29%	1.27%	0.00%
H1to6	56.25%	10.42%	9.23%
Bikes	Original	Proposed	With Noise
H1to2	63.10%	64.84%	61.76%
H1to3	60.48%	64.23%	60.00%
H1to4	53.06%	61.21%	58.39%
H1to5	47.06%	59.85%	58.45%
H1to6	35.81%	54.01%	56.23%
Boat	Original	Proposed	With Noise
H1to2	55.88%	59.79%	62.20%
H1to3	60.80%	4.01%	4.40%
H1to4	54.48%	24.20%	39.07%
H1to5	50.00%	39.64%	53.74%
H1to6	19.44%	2.77%	14.14%
Graf	Original	Proposed	With Noise
H1to2	56.20%	50.65%	60.25%
H1to3	47.46%	18.16%	23.77%
H1to4	20.14%	5.86%	11.21%
H1to5	0.00%	0.00%	0.00%
H1to6	0.00%	0.00%	0.00%
Leuven	Original	Proposed	With Noise
H1to2	62.60%	65.14%	67.47%
H1to3	56.05%	65.74%	65.00%
H1to4	57.07%	65.91%	69.28%
H1to5	58.06%	65.95%	68.16%
H1to6	47.83%	62.06%	65.77%
Trees	Original	Proposed	With Noise
H1to2	42.32%	54.40%	57.62%
H1to3	39.16%	55.11%	56.33%
H1to4	42.68%	52.25%	56.18%
H1to5	47.30%	52.48%	60.19%
H1to6	42.46%	53.29%	59.90%
UBC	Original	Proposed	With Noise
H1to2	78.81%	78.55%	65.96%
H1to3	63.36%	69.84%	64.13%
H1to4	59.10%	61.16%	60.63%
H1to5	45.21%	52.73%	57.05%
H1to6	38.36%	47.67%	49.27%
Wall	Original	Proposed	With Noise
H1to2	47.78%	60.22%	60.68%
H1to3	39.39%	59.03%	57.78%
H1to4	35.14%	42.00%	42.92%
H1to5	22.14%	33.13%	31.52%
H1to6	4.76%	16.88%	14.58%
Average	46.23%	43.64%	45.43%

and 5 bottom) two columns are connected together, so the EN_C signal is activated. After the charge redistribution is performed, the floating diffusion voltage has to be sampled, which is done by activating the row select signal, SEL. In the case of the pixels connected in the same column, two select signals are necessary to read each floating diffusion. For the pixels connected in the same row, only one select is necessary.

The floating diffusion voltages are represented with the solid and dashed lines. Since we consider an ideal switch for the transfer gate, there is no error introduced when TX is turned on or off. When the EN signal is activated the floating diffusion voltages converge to the same value, ideally equal to 1.5 V. The maximum voltage error when the row select signal is activated in the schematic simulation is of 2%. In the layout simulation, since parasitic elements that where not considered before are being added, the error increases to 5%. As an example, branch resistances and node capacitances depend on metal parameters that where not considered before. Capacitances between metal lines, also not considered in the schematic, can generate clock feedthrough, which will influence the charge sharing result whenever a switch opens or closes.

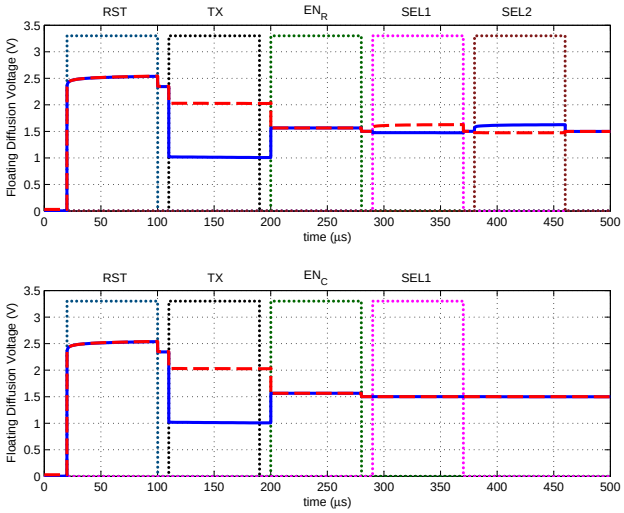

Figure 4. *Schematic simulation results for two pixels from the same column (top) and from the same row (bottom). The floating diffusion voltages are shown in solid and dashed lines, and the control signals are shown in dotted lines.*

Uniform random error was also included in system level simulations with the goal of understanding the impact that the error introduced by the hardware causes on key-point search. This error was added for every iteration of the algorithm after each average computation. After charge redistribution, when the switch EN opens and the operation is complete, the pixels used for the operation should, ideally, have the same value. From Figure 5 (top), it can be observed that there is an undesirable difference between the two pixels from different rows the switch EN_R opens. In order to take that difference into account in simulations at the system level, two error ranges where considered for a same 2×2 pixel block after the average has been computed: from 0 to 2% for two pixels in the same row inside the block and from 2% to 8% for the other two pixels. The simulations consider an error that is higher than it was observed in the simulations as a pessimistic approxi-

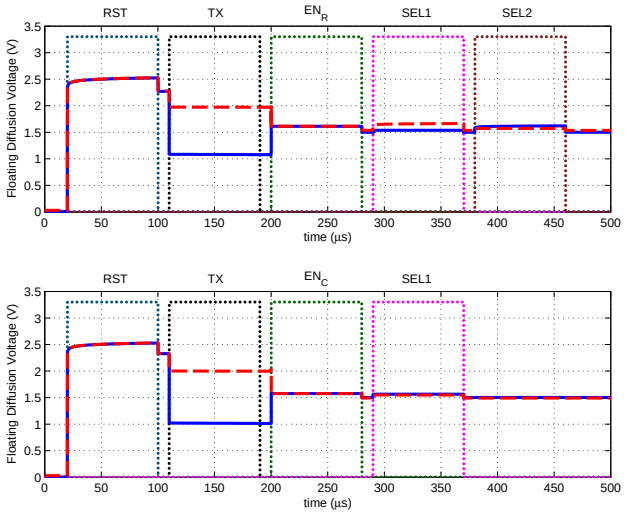

Figure 5. *Extracted layout simulation results for two pixels from the same column (top) and from the same row (bottom). The floating diffusion voltages are shown in solid and dashed lines, and the control signals are shown in dotted lines.*

mation to our problem.

As can be seen in the 'System Level Simulation Results' table, in the column 'Proposed Method with Noise' the repeatability is similar to the repeatability without noise. In some cases, the noise benefits the repeatability, bringing to evidence singularities that where not considered before, which results in a little increase in the average repeatability. On the other hand, it also increases the number of key-points found. By changing the threshold voltage, it is possible to control this increase.

Conclusion

A six-transistor pixel architecture was presented and contextualized for the generation of a scale-space data structure for the SIFT algorithm. With this architecture it is possible to perform charge redistribution among neighboring pixels, which allows the computation of an instrumental image processing task, namely the Gaussian filtering, without significantly affecting the pixel fill-factor. The new architecture helps saving computational and power resources demanded by the Gaussian filtering operation. Studies are being made with the goal of quantifying the savings provided by the proposed method.

Acknowledgments

This work has been funded partially by Brazilian research agencies (projects CNPq 204382/2014-9, CNPq 309148/2013-8, CNPq 479437/2013-0, FAPERJ E-26/201.514/2014, and FAPERJ E-26/110.099/2013), and partially by the Spanish Government through projects TEC2012-38921- C02 MINECO (European Region Development Fund, ERDF/FEDER), Junta de Andalucía through project TIC 2338-2013 CEICE, and the Office of Naval Research (USA) N000141410355.

References

[1] Á. Zarándy, Focal-Plane Sensor-Processor Chips, Springer-Verlag New York, 2011.

[2] Á. Rodríguez-Vázquez , R. Domínguez-Castro, F. Jiménez-Garrido, et al "A CMOS vision system on-chip with multi-core, cellular sensory-processing front-end", Chapter 6 in Cellular Nanoscale Sensory Wave Computers (edited by C. Baatar, W. Porod and T. Roska), Springer, 2010.

[3] R. C. González and R. E. Woods, Digital Image Processing, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[4] J. Fernández-Berni, R. Carmona-Galán and L. Carranza-González, "FLIP-Q: A QCIF Resolution Focal-Plane Array for Low-Power Image Processing", IEEE J. of Solid-State Circuits, vol. 46, No. 3, pp. 669-680, 2011.

[5] R. Carmona-Galán, J. Fernández-Berni and Á. Rodríguez-Vázquez, "Automatic DR and Spatial Sampling Rate Adaptation for Secure and Privacy-Aware ROI Tracking Based on Focal-Plane Image Processing", Image Sensor Workshop, 2015.

[6] J. Fernández-Berni, R. Carmona-Galán and Á. Rodríguez-Vázquez, "Image filtering by reduced kernels exploiting kernel structure and focal-plane averaging", IEEE European Conf. on Circuit Theory and Design (ECCTD), Linkoping, Sweden, pp. 229232, 2011.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, vol. 60, No. 2, pp. 91110, 2004.

[8] P. Viola and M. Jones, "Robust real-time face detection", International Journal of Computer Vision, vol. 57, No. 2, pp. 137154, 2004.

[9] E. R. Fossum and D. R. Hondongwa, "A Review of the Pinned Photodiode for CCD and CMOS Image Sensors", IEEE Journal of the Electron Devices Society, vol. 2, No. 3, pp. 33-43, 2014.

[10] V. Goiffon, M. Estribeau, J. Michelot, et al,"Pixel Level Characterization of Pinned Photodiode and Transfer Gate Physical Parameters in CMOS Image Sensors", IEEE Journal of the Electron Devices Society, vol. 2, No. 4, pp. 65-76, 2014.

[11] Silvaco Website [online]. Available: http://www.silvaco.com/examples/tcad/section25/example5/index.html (accessed on January 15, 2016)

[12] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features", Proc. ECCV, vol. 3951, pp 404-417, 2006.

[13] E. Parra-Barrero, J. Fernández-Berni, F. D. V. R. Oliveira, et al, "High-Level Performance Evaluation of Object Detection Based on Massively Parallel Focal-Plane Acceleration Requiring Minimum Pixel Area Overhead", Proc. VISAPP (accepted paper), 2016.

[14] C. Schmid, R. Mohr and C. Bauckhage, "Evaluation of Interest Point Detectors", International Journal of Computer Vision, vol. 37, No. 2, pp. 151-172, 2000.

[15] OpenCV Website [online]. Available: http://docs.opencv.org/modules/nonfree/doc/feature_detection.html (accessed on January 15, 2016)

[16] OpenCV's SIFT implementation [online]. Available: https://gist.github.com/lxcxx/7088609 (accessed on January 15, 2016)

[17] Collaborative work between: the Visual Geometry Group, Katholieke Universiteit Leuven, Inria Rhone-Alpes and the Center for Machine Perception. Website [online]. Available: http://www.robots.ox.ac.uk/∼vgg/research/affine/

[18] K. Mikolajczyk, T. Tuytelaars, C. Schmid, et al, "A comparison of affine region detectors", International Journal of Computer Vision, vol. 65, No. 1, pp. 43-72, 2005.

[19] M. Suárez-Cambre, "Low power CMOS vision sensors for scale and rotation invariant feature detectors using CMOS heterogeneous smart pixel architectures", Ph.D. Thesis, University of Santiago de Compostela, 2015.

[20] R. Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited, 2010.

Author Biography

Fernanda Duarte Vilela Reis de Oliveira *graduated as an Electronic Engineer in 2012 and by the end of 2013 received her M.S. degree in Electric Engineering, both from the Federal University of Rio de Janeiro (UFRJ), Brazil. She is currently pursuing her Ph.D. degree in microelectronics at the Electrical Engineering Program of COPPE/UFRJ. She did a one year internship in the Microelectronics Institute of Seville in 2015. Her research fields are image sensors and image processing.*

José Gabriel Rodríguez Carneiro Gomes *graduated in Electrical Engineering from the Federal University of Rio de Janeiro in 1999 (magna cum laude). He obtained M.S. degrees in Electrical Engineering from COPPE/UFRJ (2000) and from the University of California at Santa Barbara (UCSB, 2003), and a Ph.D. degree in Electrical Engineering from UCSB (2004). In 2005, he was a post-doctoral researcher with the Electrical Engineering Program at COPPE/UFRJ. In 2006, he joined the faculty of the Electronics and Computer Engineering Department at the Federal University of Rio de Janeiro, where he is currently an Associate Professor. Since 2007, he is also part of the faculty at the Electrical Engineering Program at COPPE/UFRJ. His professional experience concentrates on Electronics Instrumentation, with an emphasis on CMOS image sensors, image compression, and neural networks. He received "Young Researcher of Our State" research grants from FAPERJ/Brazil for terms 2009/2012 and 2015/2017. He and his co-authors received the Best Paper Award at the 25th Symposium on Integrated Circuits and Systems Design (SBCCI 2012) in Brasilia, Brazil. He was a recipient of the IEEE Circuits and Systems Society Chapter-of-the-Year Award (Region 9, 2009). He is an IEEE member since 2001.*

Ricardo Carmona-Galán *(M'04) received the Licenciado and Ph.D. degrees in physics, in the speciality of electronics, from the University of Seville, Spain, in 1993 and 2002, respectively. He was a Research Assistant with the University of California, Berkeley. From 1999 to 2005, he was an Assistant Professor with the Department of Electronics and Electromagnetism, School of Engineering, University of Seville. Since 2005, he holds a tenured position with the Institute of Microelectronics of Seville (CSIC). He also held a post-doctoral position at the University of Notre Dame, IN, USA (2006-2007). His main research areas are vision chips, in particular, smart CMOS imagers for low-power vision applications like robotics, vehicle navigation, and vision-enabled wireless sensor networks. He is also interested in CMOS-compatible sensing structures for LWIR and MWIR imaging, single-photon detection, and detector for X-ray and highenergy physics, and also in the implementation of high resolution smart imagers in 3-D integrated circuit technologies. He is a member of the IEEECASS Technical Committees on Cellular Nanoscale Networks and Array Computing and on Sensory Systems. He is member of the Steering Committee of the Workshop on Architecture of Smart Cameras, and has recently chaired the 9th International Conference on Distributed Smart Cameras in-cooperation with ACM-SIGBED. He has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMSI: REGULAR PAPERS from 2012-2013*

Jorge Fernández-Berni *has co-authored over 50 papers related to vision chips and embedded vision systems in refereed journals, conferences and workshops. He has been a visiting researcher in the Cellular Sensory and Optical Wave Computing Laboratory (SZTAKI, Budapest, Hungary), the Image Processing and Interpretation group (TELIN-IPI, Ghent University, Belgium) and the Center for Nanoscience and Technology, NDnano (University of Notre-Dame, IN USA).*

Ángel Rodríguez-Vázquez *(M'80-F'96) received the Licenciado and the Ph.D. degrees in físcia electrónica from the University of Seville, Seville, Spain, in 1977 and 1983, respectively. He is currently a Full Professor of Electronics with the University of Seville. His research is on the design of analog and mixedsignal front-ends for sensing and communication, including smart imagers, vision chips and low-power sensory-processing microsystems. He has authored 11 books, 34 additional book chapters, and some 150 journal articles in peer-review specialized publications. He has presented invited plenary lectures at different international conferences and has received a number of awards for his research (the IEEE Guillemin-Cauer Best Paper Award, two Wileys IJCTA Best Paper Awards, two IEEE ECCTD Best Paper Awards, one SPIE-IST Electronic Imaging Best Paper Award, the IEEE ISCAS Best Demo-Paper Award, and the IEEE ICECS Best Demo-Paper Award). He was elected as a Fellow of the IEEE for his contributions to the design of chaos-based communication chips and neuro-fuzzy chips. His research work received some 6,800 citations; he has an h-index of 43 and an i10-index of 134. He has always been looking for the balance between long-term research and innovative industrial developments. He founded AnaFocus Ltd., in 2001, on the basis of his patents on vision chips, and served as CEO, on leave from the University, until 2009, when the company reached maturity as a worldwide provider of smart CMOS imagers and vision systems-on-chip. He has served as an Editor, an Associate Editor, and a Guest Editor of different IEEE and non-IEEE journals, is in the committee of several international journals and conferences, and has been the Chair of several international IEEE and SPIE Conferences. He served as VP Region 8 of the IEEE Circuits and Systems Society (2009-2012) and as the Chair of the IEEE CASS Fellow Evaluation Committee (2010, 2012, 2013, 2014, and 2015).*